

Path Planning for Mobile Robots Using a Video Camera Network

Adam Hoover

Electrical and Computer Engineering Dept.
Visual Computing Lab
University of California, San Diego
La Jolla, CA 92093-0407
hoover@vision.ucsd.edu

Bent David Olsen

Dept. of Medical Inform. & Image Analysis
Laboratory of Image Analysis
Aalborg University
Denmark
einstein@vision.auc.dk

Abstract

We describe a novel method to plan paths for mobile robots in an environment observed by a network of video cameras. The multiple video streams are fused to generate a spatial-temporal map of freespace [8]. Updated at 5 Hz, this map concurrently locates both static and dynamic obstacles. Our novel method computes a polynomial-shaped path from current position to goal position for each map update (at 5 Hz), thus accomplishing both path planning and dynamic obstacle avoidance simultaneously. We demonstrate our methods operating in several dynamic scenarios, and compare our methods against paths planned using a numeric potential field.

1 Introduction

The two classical motivations for robotics are exploration and automation. Robots constructed for exploration require onboard sensors and processing capabilities in order to bring operability to an unknown environment. Robots constructed for automation can take advantage of workcells, devoid of human or other mechanical presence, eliminating the need for environment sensing. In this paper we describe an alternative mobile robotic system.

The central theme is use off-board sensors, located in the environment. Figure 1 shows an example sensor network consisting of four video cameras. The idea is that mobile robots working in the area tune in to broadcasts from the video camera network (or from environment-based computers processing the video frames) to receive sensor data. This *distributed sensing* configuration offers several advantages over onboard sensing:

1. Data fusion over space and time is simpler using stationary sensors. In [8], we describe an al-

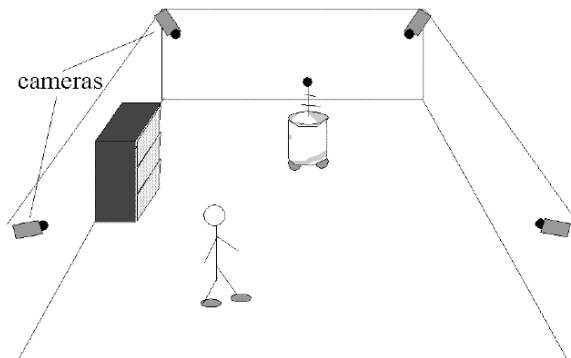


Figure 1: A video camera network.

gorithm to compute a real-time occupancy map from a video network like the one in Figure 1. This system is reviewed in Section 2.

2. The combined field-of-view of a sensor network is larger than what can be provided by onboard sensors. Using onboard sensors, a robot has only *line-of-sight sensing*, and so must rely upon partial path-planning strategies (see for instance [13, 14]). In Section 3 we describe a novel path-planning strategy that takes advantage of distributed sensing.
3. The component of image blur caused by camera motion is eliminated. This problem may also be overcome by navigating in a move-stop-image cycle, at the cost of operating in a static environment (see for instance [1, 2]).

Our novel algorithm considers paths fitted to polynomial curves. The assumption is that polynomial-shaped paths are frequently sufficient for navigating common indoor environments. A major point is that polynomial paths may also be computed at rates sufficient to accommodate dynamic obstacle avoidance.

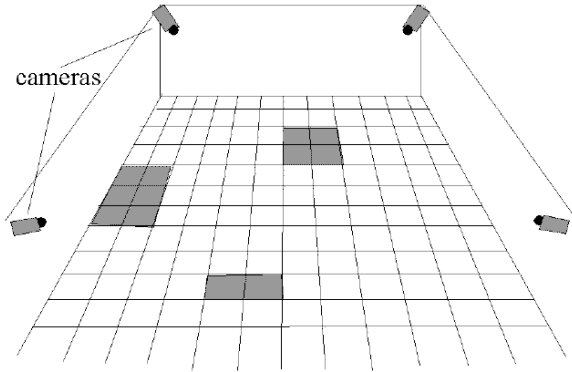


Figure 2: A spatial occupancy map.

We compare these methods against paths planned using a numerical potential field [3], and demonstrate our methods navigating a mobile robot in several dynamic situations.

2 Distributed Sensing

For this work we constructed a network of video cameras similar to the depiction in Figure 1. The camera layout resembles a security video network. The cameras are all connected to a single computer that processes the video feeds to produce a spatial-temporal occupancy map [8]. The occupancy map is a two-dimensional raster image, uniformly distributed in the floor-plane. Each map pixel contains a binary value, signifying whether the designated floorspace is empty or occupied. Figure 2 shows an example occupancy map where grey cells indicate the space is occupied and white cells indicate the space is empty. A spatial frame of the occupancy map is computed from a set of intensity images, one per camera, captured simultaneously. Temporally, a new map frame can be computed on each new video frame sync signal. Thus in effect, the map is itself a video signal, where the pixel values denote spatial-temporal occupancy. Figure 3 shows an example occupancy map frame from our system and a concurrent view of the action in the room. The chairs being pushed across the room are visible in the map as star-shaped areas, denoting the occupancy of the chair bases. The person standing at the right of the image is just visible at the bottom of the occupancy map.

3 Path Planning

Dynamic obstacle avoidance and static goal navigation may be viewed as two separate problems. The former may be accomplished using onboard sensors, the latter by using a complete map of the static environment. A multi-layered approach to navigation

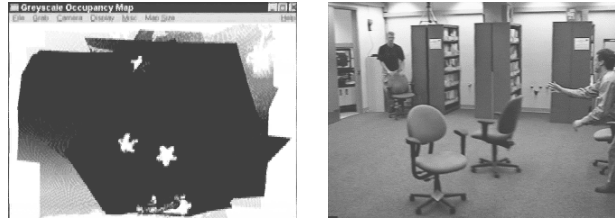


Figure 3: An example occupancy map frame, and a concurrent view of the action in the room.

accomplishes dynamic obstacle avoidance locally while navigating towards a static goal [5]. Methods for static goal navigation include those based upon a visibility graph [15], the Voronoi diagram [17], cell decomposition [4], a potential field [3], and the configuration space [11]. A review of many of these approaches may be found in [12].

Algorithms using occupancy maps have been demonstrated for robot path planning, collision determination, and other navigation and location-related tasks (see for instance [6, 10, 18]). These works consider the problems from the point of view of a robot using on-board sensors or a priori maps for navigation. Two properties of our distributed sensing system provide for an alternative solution:

1. The distributed sensor network allows path computation from start to goal point regardless of the robot's current field-of-view.
2. The real-time occupancy map captures both static and dynamic objects concurrently.

If a new start-to-goal path can be planned for each occupancy map frame then path planning and dynamic collision avoidance are solved simultaneously.¹ In Section 3.1 we describe a novel path planning algorithm designed to meet this timing requirement. In Section 3.2 we describe our implementation of a well-known path planning algorithm based upon a numerical potential field. In Section 3.3 we discuss the differences between these methods, which are highlighted via experimentation in Section 4.

3.1 Polynomial Path Planner

The goal of the following algorithm is, given an occupancy map snapshot, search a defined collection of

¹Being able to compute trajectories for dynamic objects could theoretically improve collision avoidance [7]. However, animate (alive) dynamic objects rarely follow predictable trajectories for very long. Consider two people approaching each other in a narrow corridor. As the bodies meet, there is often a short period of dancing back-and-forth to avoid collision. The method described here imitates this approach to navigation.

polynomials connecting the start and goal points, reporting the shortest collision-free path. Let $E[x_i, y_j]$, $1 \leq i \leq X$, $1 \leq j \leq Y$, i and j integers, define a model of two-dimensional space. Let $M[x_i, y_j]$ describe an occupancy map of space E , where each $M[x_i, y_j]$ models a rectangle of space $\frac{1}{X} \times \frac{1}{Y}$ units in size. In this occupancy map, $M[x_i, y_j] = 0$ signifies a rectangle of freespace while $M[x_i, y_j] \neq 0$ signifies a rectangle of (at least partially) occupied space. Let $S = (x_s, y_s)$ define a starting position, $G = (x_g, y_g)$ define a goal position, and R be the distance that encloses the maximum radius of the rigid body which is to travel from S to G . A path from S to G is defined as

$$\begin{aligned}
P &= \{(p_{x_1}, p_{y_1}), (p_{x_2}, p_{y_2}), \dots, (p_{x_N}, p_{y_N})\}, \\
\text{such that } (p_{x_1}, p_{y_1}) &= S, \quad (p_{x_N}, p_{y_N}) = G, \\
-1 &\leq (p_{x_i} - p_{x_{i-1}}) \leq 1 \quad \text{and} \\
-1 &\leq (p_{y_i} - p_{y_{i-1}}) \leq 1 \quad \forall i = 2 \dots N, \\
(p_{x_i}, p_{y_i}) &\neq (p_{x_j}, p_{y_j}) \quad \forall i, j = 1 \dots N, \quad i \neq j \quad (1)
\end{aligned}$$

This definition describes any non-self-intersecting path that connects S to G . Obviously, many of these paths will be unreasonable, for instance having a jagged or wandering shape. A reasonable path should be smooth, implying that it should be well-modeled by a continuously differentiable curve $F(x(t), y(t))$ where $F(x(0), y(0)) = S$ and $F(x(1), y(1)) = G$. For such a path, we may define a function $F^\perp(x(t, u), y(t, u))$ such that

$$\begin{aligned}
F^\perp(x(t, u), y(t, u)) \cdot F'(x(t), y(t)) &= 0, \\
\|(F^\perp(x(t, 0), y(t, 0)) - F^\perp(x(t, \frac{1}{2}), y(t, \frac{1}{2})))\| &= R, \\
\|(F^\perp(x(t, \frac{1}{2}), y(t, \frac{1}{2})) - F^\perp(x(t, 1), y(t, 1)))\| &= R, \\
F^\perp(x(t, \frac{1}{2}), y(t, \frac{1}{2})) &= F(x(t), y(t)) \quad (2)
\end{aligned}$$

Equation (2) models a line of width $2R$, centered on and perpendicular to $F(x, y)$, along the length of F . A path P is valid (traversable by the robot) if

$$\int_{t=0}^1 \int_{u=0}^1 M[F^\perp(x(t, u), y(t, u))] = 0 \quad (3)$$

Equation (3) tests a one-dimensional line of free-space at each path point. This is more efficient than testing a two-dimensional shape (e.g., the shape of a rigid body) at each path point, because it reduces the potential redundant overlap from tests of neighboring path points.

We define a set of N_l second-order polynomial paths from S to G as follows:

$$\begin{aligned}
F_l(x(0), y(0)) &= S, \quad F_l(x(1), y(1)) = G, \\
F_l(x(\frac{1}{2}), y(\frac{1}{2})) &= S + \frac{\|S-G\|}{2} (\cos l\theta, \sin l\theta), \\
l &= \frac{-N_l}{2} \dots \frac{N_l}{2} \quad (4)
\end{aligned}$$



Figure 4: A set of polynomial paths (left figure) from start (upper left) to goal. The best path (right figure) is the shortest collision-free path in the set.

Each polynomial is defined by three points: S , G , and a set of points equi-angular from S at a radius of $\frac{\|S-G\|}{2}$. The values for the number of polynomials N_l and the equi-angular step θ control the coverage of the set. Figure 4 shows an example set of polynomial paths for $N_l = 13$ and $\theta = 15^\circ$ for points S and G at a distance of roughly 400 map pixels, and the best path in the set.

The following pseudocode finds the shortest collision-free polynomial path F_l from the set of polynomial paths defined in Equation (4):

```

given  $S, G$ 
loop  $l = 0, -1, 1, -2, 2, \dots, \frac{-N_l}{2}, \frac{N_l}{2}$ 
  solve equation (4) for  $F_l(x(\frac{1}{2}), y(\frac{1}{2}))$ 
  solve for points on  $F_l$ 
  let PATH_OK = TRUE
  loop  $i=0 \dots N-1$  (number of points on  $F_l$ )
    solve for points on  $F_l^\perp$ 
    solve equation (3) for  $M[F^\perp(x(i, u), y(i, u))]$ 
    if  $M \neq 0$  PATH_OK = FALSE, exit loop
  end loop
  if PATH_OK = TRUE exit loop
end loop

```

Solution details may be found in [16].

3.2 Potential Field Path Planner

Potential field methods represent the robot as a particle under the influence of an artificial potential field, whose local variations are expected to reflect the structure of the free space. The potential function is typically defined over free space as the sum of an attractive potential, pulling the robot toward the goal configuration, and a repulsive potential, pushing the robot away from obstacles. The numerical potential field path planning method [3] is a hybrid method between a skeleton method and a traditional potential field method.

The planner can be divided into four steps:

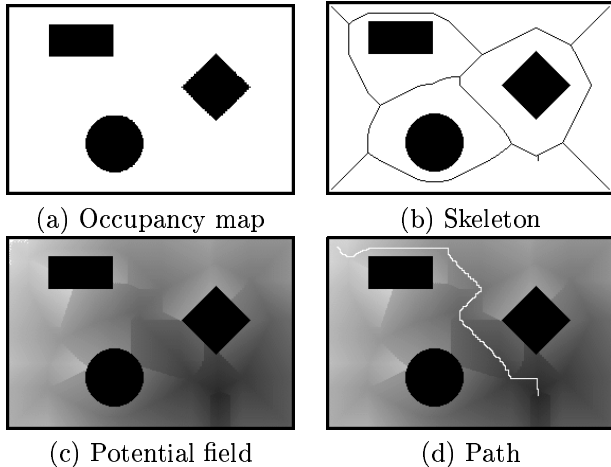


Figure 5: The numerical potential field method first computes a skeleton (b), then a numerical potential field (c) in which an A* search algorithm can be used to find a path (d).

1. Compute the Voronoi diagram of the free space, thereby generating a skeleton.
2. Compute a potential field in the skeleton.
3. Compute a numerical potential field outside the skeleton based on the values of the numerical potential field in the skeleton.
4. Connect the goal and start position by (a) connecting the goal to the skeleton, and (b) performing a best first search in the augmented skeleton.

Figure 5(b) shows an example of a skeleton generated using this method. Figure 5(c) shows the corresponding numerical potential field. An A* search algorithm on this augmented skeleton produces the sample path shown in Figure 5(d). The details of this method may be found in [3].

3.3 Discussion

The polynomial path planning algorithm was developed based upon the assumption that a mobile robot, using distributed sensing, is to navigate common indoor floorplans. For instance, the environment may be a set of rooms connected via doorways and hallways, containing some furniture, with sensible portions of the floorspace reserved for walking. In these environments, many reasonable paths for navigation may be modeled by two-dimensional polynomials in the plane of the floor. The emphasis of this algorithm is on speed and smoothness. In contrast, the numerical field potential path planner is capable of locating a path given

any configuration. The emphasis of this algorithm is on guaranteeing a solution.

In relatively uncluttered areas, the numerical potential field path planner tends to produce poor solutions. Even though a straighter path may be available, local minima in the potential field can cause wide deviations from a straighter path (for instance, see the path in Figure 5(d)). In contrast, the polynomial path planner always finds a smoothly continuous solution in open areas. In cluttered or maze-like areas, the numerical potential field path planner shows its strength, by always locating a solution if one exists. In contrast, the polynomial path planner will usually fail to find a solution in these situations.

In our prototype system the polynomial path planner runs at better than two orders of magnitude faster than the potential field path planner, on the same data and hardware. It is interesting to consider a strategy involving both methods, since they make such strong complements. This is a topic of current work.

4 Experiments

We demonstrate our methods using the following system. The experimental room is 6.5×9 sq. m. in area. Bookshelves line most of the walls. Four Sony XC-999 CCD cameras, two with 3.5 mm lenses and two with 6.0 mm lenses, are mounted on top of the bookshelves. The cameras are connected to a single computer, equipped with two Matrox Meteor digitizers, a Matrox Millennium II graphics display, and a 233 MHz Pentium processor. The images are processed at full resolution (480×640) to produce a map of size 480×640 pixels. This system achieves an occupancy map frame rate of approximately 5 Hz. The map is broadcast to other computers by displaying it via the standard RGB display and using an RGB-to-NTSC converter attached to the graphics display. The NTSC signal is received by a Silicon Graphics O2 workstations for path planning. The mobile robots receive motor control commands from the SGI workstations via RF links. The control loop is closed by tracking the robot's position in the occupancy map. Further details, including the control and tracking algorithms, may be found in [16].

The following experiments were designed to demonstrate "real-world" situations in which a mobile robot may have to navigate around people and other dynamic obstacles in an indoor environment. During these experiments, the polynomial path planner operated at 5 Hz, computing a new path for each map frame. In each case the robot maintained an average speed of approximately 20 cm/sec, which is near the robot's maximum speed. Some demonstrations may

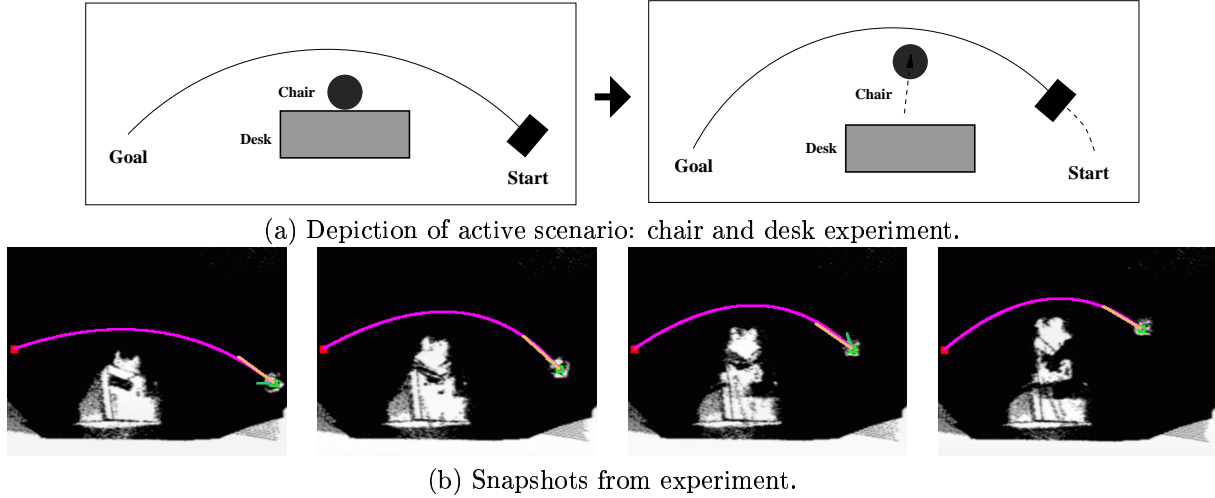


Figure 6: As the robot moves to the opposite side of a person sitting at a desk, the person pushes back in the chair. Navigation proceeds smoothly as the path planner continually plans a wider path around the obstacles.

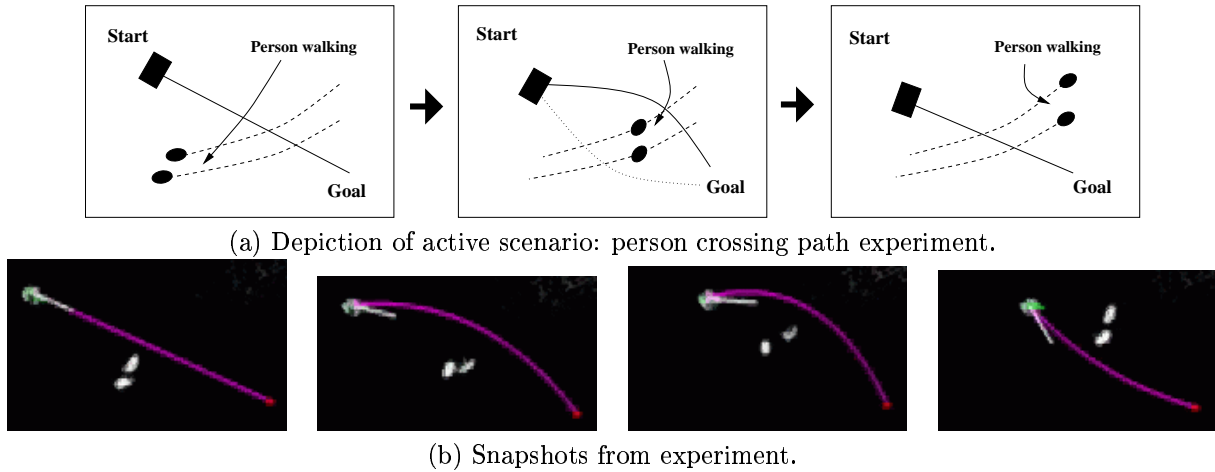


Figure 7: A person crossing the robot's path makes the path planner find paths on both sides (first one side, then the other side) of the obstacle as it passes. Forward progress is continually maintained.

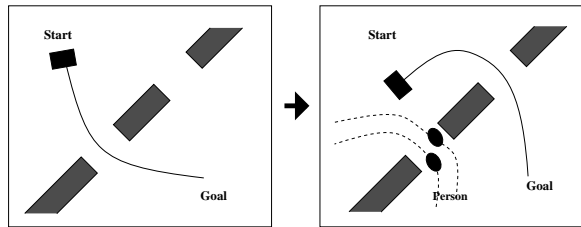
be viewed on video in [9].

Figure 6 shows a scenario in which a robot is to reach a goal point on the opposite side of a person sitting at a desk. While the robot is in motion, the person pushes back in the chair and blocks the original path. By computing a new polynomial path at the frame rate of the occupancy map signal, the robot is able to react to the dynamic situation and smoothly and continuously navigate a wider path.

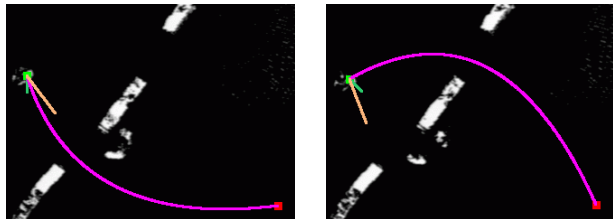
Figure 7 shows a scenario in which a person crosses the robot's path while it is navigating towards a goal point. While the person crosses the robot's path, the side of the obstacle (person) which gives the best path is momentarily in question. This causes a slight back-

and-forth (while still making forward progress) motion until the straight-line path is cleared.

Figure 8 shows a scenario in which a robot is to reach a goal point on the opposite side of a wall containing two doorways. Once motion has begun and navigation has proceeded towards a particular doorway, a person moves into that doorway and blocks it, causing the robot to navigate through the other doorway. Figure 4(b) shows the polynomial path planner operating in this scenario. Figure 4(c) shows the numerical potential field path planner operating in this scenario. The potential field path planner operated at approximately 1 Hz during this run, by downsampling the occupancy map by a factor of sixteen.



(a) Depiction of action: blocked doorway experiment.



(b) Snapshots using polynomial path planner.



(c) Snapshots using potential field path planner.

Figure 8: When the doorway through which the robot was planning to navigate gets blocked, it immediately navigates towards the other doorway.

5 Acknowledgment

The authors thank Dr. Ramesh Jain for graciously supporting this work in the Visual Computing Lab at the University of California, San Diego.

References

- [1] C. S. Andersen, et. al., "Navigation Using Range Images on a Mobile Robot", in *Robotics and Autonomous Systems*, **10**, 1992, pp. 147-160.
- [2] M. Asada, et. al., "Dynamic Integration of Height Maps into a 3D World Representation from Range Image Sequences", in *Int. J. of Computer Vision*, **9:1**, 1992, pp. 31-53.
- [3] J. Barraquand, B. Langlas and J. Latombe, "Numerical potential field techniques for robot path planning", in *IEEE Trans. on Systems, Man and Cybernetics*, vol. 22, 1992, pp. 224-241.
- [4] R. A. Brooks, "Solving the Find-Path Problem by Good Representation of Free Space", in *IEEE T-SMC*, **3:13**, 1983, pp. 190-197.
- [5] R. A. Brooks, "A Robust Layered Control System for a Mobile Robot", in *IEEE Trans. on Robotics & Automation*, **2:1**, April 1986, pp. 14-23.
- [6] A. Elfes, "Using occupancy grids for mobile robot perception and navigation", in *IEEE Computer*, **22:6**, June 1989, pp. 46-58.
- [7] K. Fujimura and H. Samet, "A Hierarchical Strategy for Path Planning Among Moving Obstacles", in *IEEE T-RA*, **5:1**, Feb 1989, pp. 61-69.
- [8] A. Hoover and B. Olsen, "A Real-Time Occupancy Map from Multiple Video Streams", in *IEEE ICRA*, 1999, pp. 2261-2266.
- [9] A. Hoover and B. Olsen, "Sensor Networked Robotics", in the video proc. of *IEEE Int'l Conf. on Robotics & Automation*, Detroit, MI, 1999.
- [10] Y. Hwang and N. Ahuja, "Gross Motion Planning - A Survey", in *ACM Computing Surveys*, **24:3**, Sep. 1992, pp. 219-291.
- [11] J. Ilari and C. Torras, "2D Path Planning: A Configuration Space Heuristic Approach", in *Int'l J. of Robotics Research*, **9:1**, Feb 1990, pp. 75-91.
- [12] J.-C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Second Edition, 1991.
- [13] V. Lumelsky, S. Mukhopadhyay and K. Sun, "Dynamic path planning in a sensor-based terrain acquisition", in *IEEE T-RA*, **6:4**, 1990, pp. 462-472.
- [14] V. Lumelsky and T. Skewis, "Incorporating range sensing in the robot navigation function", in *IEEE T-SMC*, **20:5**, 1990, pp. 1058-1069.
- [15] M. Montgomery, D. Gaw and A. Meystel, "Navigation algorithm for a nested hierarchical system of robot path planning among polyhedral obstacles", in *IEEE ICRA*, 1987, pp. 1612-1622.
- [16] B. D. Olsen, "Robot Navigation Using a Sensor Network", Master's Thesis, Laboratory of Image Analysis, Aalborg University, June 1998.
- [17] O. Takahashi and R. Schilling, "Motion Planning in a Plane Using Generalized Voronoi Diagrams", in *IEEE T-RA*, **5:2**, April 1989, pp. 143-150.
- [18] R. Talluri and J. K. Aggarwal, "Position Estimation Techniques for an Autonomous Mobile Robot - A Review", in *Handbook of Pattern Recognition and Computer Vision*, C. Chen, L. Pau and P. Wang (ed.), 1993, pp. 769-801.